

EPICS data archiver at SSRF beamlines

HU Zheng (胡正),^{1,2} MI Qing-Ru (米清茹),¹ ZHEN Li-Fang (郑丽芳),¹ and LI Zhong (黎忠)^{1,*}

¹Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China

²University of Chinese Academy of Sciences, Beijing 100049, China

(Received August 23, 2013; accepted in revised form November 26, 2013; published online March 20, 2014)

The control system of SSRF (Shanghai Synchrotron Radiation Facility) is based on EPICS (Experimental Physics and Industrial Control System). Operation data storage for synchrotron radiation facility is important for its status monitoring and analysis. At SSRF, operation data used to be index files recorded by traditional EPICS Channel Archiver. Nevertheless, index files are not suitable for long-term maintenance and difficult for data analysis. Now, RDB Channel Archiver and MySQL are used for SSRF beamline operation data archiving, so as to promote the data storage reliability and usability. By applying a new uploading mechanism to RDB Channel Archiver, its writing performance is improved. A web-based GUI (Graphics User Interface) is also developed to make it easier to access database.

Keywords: Database, RDB Channel Archiver, Markov Auto-Complete, EPICS, SSRF

DOI: 10.13538/j.1001-8042/nst.25.020103

I. INTRODUCTION

Shanghai Synchrotron Radiation Facility (SSRF) is capable of building a number of beamlines [1] and experimental stations for physical, chemical, and biological studies. For each beamline, over 200 signals that provide information of the status of vacuum, temperature, cooling water flow etc. should be monitored and archived accurately and reliably. This is also demanded for the SSRF accelerators, hence a great quantities of signals need to be archived.

At SSRF, the control system software is based on EPICS and the Channel Archiver is used to archive the operation data. The RDB Channel Archiver is a newly upgraded version of Channel Archiver. It is implemented by Java and uses relational database to provides better data storage than the original index files. MySQL, Oracle and PostgreSQL are three kinds of relational database supported by RDB Channel Archiver. Actually, RDB Channel Archiver communicates with the database by using JDBC (Java Database Connectivity) to guarantee a reliable connection, it supports any database which has JAVA API.

for a targeted PV, and the IOC holding the PV will response it and establish a channel, which enables the client to read values from the PV, write values to the PV, and monitor the PV.

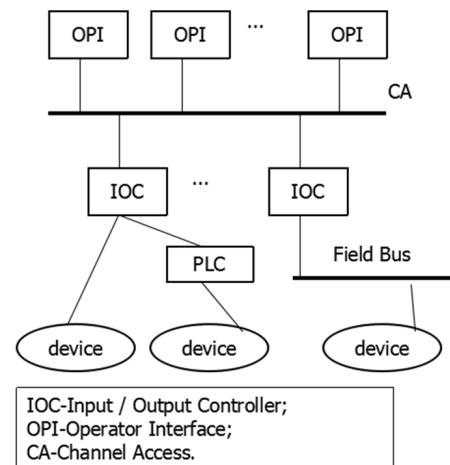


Fig. 1. EPICS framework.

II. EPICS STRUCTURE

EPICS is a set of Open Source software tools, libraries and applications. It is widely used to create distributed soft real-time control systems for scientific instruments [2]. As Fig. 1 shows, it mainly consists of three parts: IOC (Input/Output Controller), OPI (Operation Interface), and CA (Channel Access) [3]. CA is a set of communication protocols based on TCP/IP. It uses multiple clients to multiple servers access method. And an important data unit of CA is PV (Process Variable), which usually represents a signal. To establish a data access channel, its client broadcasts across the network

III. RDB CHANNEL ARCHIVER AND ITS IMPROVEMENT

A. Structure of RDB Channel Archiver

RDB Channel Archiver is implemented to make an effective data archive on SSRF beamlines. It is a major part of the CCS (Control System Studio), an open source toolset developed under Eclipse by Kay Kasemir from the Oak Ridge National Laboratory [4] of USA. With the main goal of transmitting PV data from IOC to the relational database, RDB Channel Archiver uses the JCA to connect EPICS IOCs, and JCA is a Java Channel Access protocol which works well with

* Corresponding author, lizhong@sinap.ac.cn

original Epics Channel Access [5]. Although its operation mechanism (read and store the data via periodic scanning or monitoring the PV, as shown in Fig. 2) is similar to Channel Archiver, all access to data in RDB Channel Archiver uses interface layers, and implementations are provided in separate plug-ins [6] which make it easily compatible with different databases.

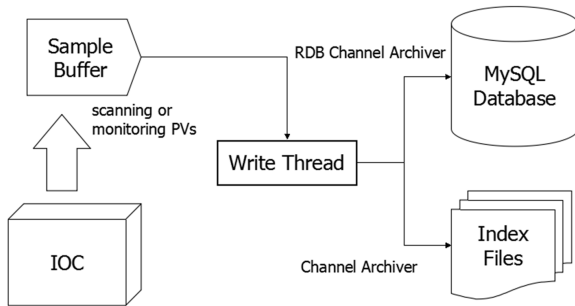


Fig. 2. Operation mechanism of Channel Archiver and RDB Channel Archiver.

RDB Channel Archiver supports Oracle, PostgreSQL and MySQL, which are all full-blown database management systems. We use MySQL as the database for SSRF beamlines, because Oracle's license is not free whilst PostgreSQL performance is not as high as MySQL. Although MySQL has some functionality shortages, this does not affect our data archiver. The E-R diagram of database is shown in Fig. 3 where the tables 'smpl_eng', 'chan_grp', and 'channel' represent the information and configuration of a PV channel, while the table 'sample' and 'array_val' hold the value of archived PVs.

B. Improvement of RDB Channel Archiver

In the original RDB Channel Archiver, a write thread is used to scan the sample buffers and send the data to the database. At SSRF, an additional write thread and an uploading controller named uploader are applied. Improvement in the operation mechanism is that the two write threads operate alternatively with the buffers, and the database is under control of the uploader, which controls the upload method for each write thread by incorporating their upload method in a java synchronized function. Therefore, no conflict occurs when two write threads are uploading simultaneously.

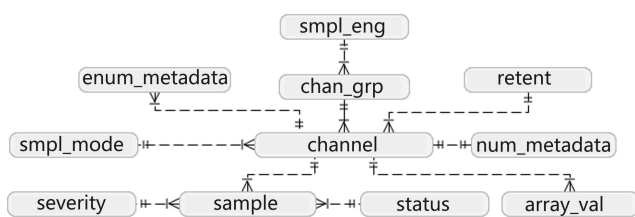


Fig. 3. (Color online) Database E-R diagram.

As shown in Fig. 4, the parts drew by solid line are original

RDB Channel Archiver components, and dashed line components are added to the RDB Channel Archiver. Symbol 'SQL' in Fig. 4 means upload SQL query and send it to the database. Considering one SQL query usually consumes more time than preparing process before executing that query, the uploading is more likely a continuous process and becomes more effective when two write threads work together.

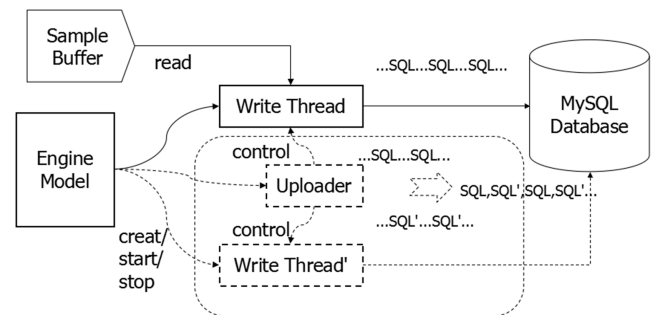


Fig. 4. Modified RDB Channel Archiver against original RDB Channel Archiver.

Nevertheless, nothing is changed to the upper layer, because launching the two upper layer plug-ins, EngineConfigImport and ArchiveEngine, stays the same [7]. The whole process remains invisible to the final user.

Upload performances of original and improved RDB Channel Archiver were tested with a set of write period and batch size, which are major parameters of the RDB Channel Archiver. The database server was IBM System x3850, with the Operating System of Red Hat Enterprise Linux Server release 5.3 (Tikanga). Simulate PVs were used to test the two RDB Channel Archivers, and the PV scanning rate was 2000 values per second. This rate approximates a high operation load at SSRF. The write speed of the improved RDB Channel Archiver was compared with the original one, using various batch sizes over the write periods of 5 s, 10 s, 15 s and 20 s, and results are shown in Fig. 5. The uploading performance is promoted more than 10% in average.

IV. WEB-BASED GUI

A. Frame work of the GUI

A web-based GUI (Graphics User Interface) was developed for users of the SSRF beamlines (Fig. 6). Users can access the database by web browser without installing any other application from any networked computer. It provides great convenience for user accessibility. Its server was deployed under LAMP structure, which works well with the MySQL database. It is totally driven by PHP, Javascript and HTML, and the chart elements are based on high charts library, which has better compatibility and speed than flash.

There are four parts in the web-based GUI. Part I is the archived PV chart. It has two sub-charts: one master chart of a large time scope to show whether there are data or not for the

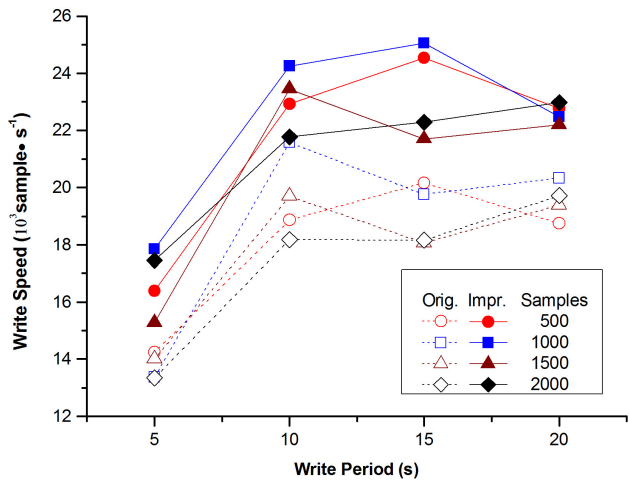


Fig. 5. (Color online) Performance comparison of write speed using various batch sizes of 500, 1000, 1500 and 2000 samples. The blank and solid data symbols stand for the original and improved RDB Channel Archivers.

operations. The exact PV data for corresponding time stamp can be shown in tooltip when user points at the detailed chart. The detailed chart could be recorded into a file and saved in desired disk location.

Part II provides live PV chart. It is redraw each second, showing the current values of a specified PV. And this chart contains 20 values which representing values of the last 20 seconds for that PV. Detention measured in millisecond is also shown as text. If the detention is over 20 minutes, the text will tell that there is no live PV data.

Part III is a PV selection area. It provides a way to select PV by attributes, such as name and region of the beamline, and monitor types. When these attributes are specified, facilities with corresponding PVs will be displayed under the options shown in Fig. 7. The users can select a certain PV by clicking one radio. All the radios are not deployed before they are generated by user input and database. Part IV is an input box for directly typing PV name, which has the same purpose as part III.

B. Markov Auto-Complete and its implement

A new kind of Auto-Complete (or suggest) is applied for the input box in the GUI. It is different from usual Auto-Complete which does not give a set of suggested words once the users input a few letters. Instead, it only gives a set of suggested fields of all the possible words each time until all the fields compose the complete suggestion word. For example, if the target PV is “X14W:FE:FM1:TC1:AI”, the users need to type in “X” and select “X14W:”, then select “X14W:FE:”, “X14W:FE:FM1:”, “X14W:FE:FM1:TC1:AI” to get the target PV. All the user selections are automatically displayed in the input box, and the users need only key ‘up’ or ‘down’ and ‘enter’ to complete the selection. This process is shown in Fig. 8. This can hardly be done by usual Auto-Complete if there are too many suggestion words start with “X”, indicating advantage of this method. In fact, the whole word is divided into several parts, and is treated as a markov chain.

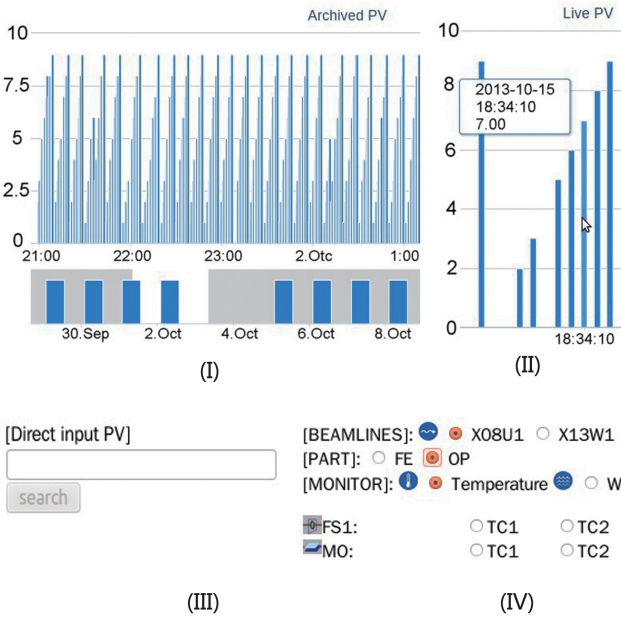


Fig. 6. (Color online) Data retrieval web page.

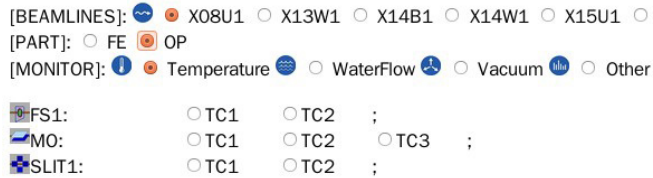


Fig. 7. (Color online) PV selected by attributes.

specified PV, and another to show details of the selected time span from master chart and to provide basic zoom and pan

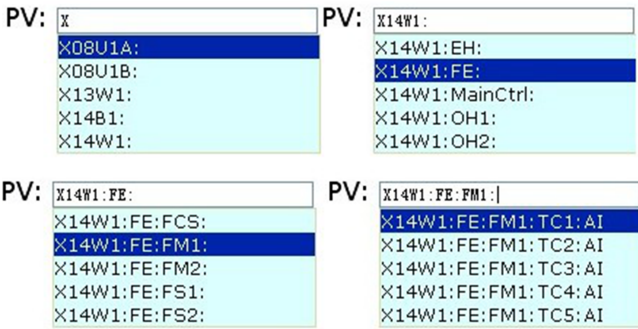


Fig. 8. (Color online) Process of selecting PV by ‘Markov Auto-Complete’.

V. CONCLUSION

RDB Channel Archiver is a promising upgrade for Channel Archiver. It offers a mature solution for long term operation data storage, being easier to access. Upload mechanism

of RDB Channel Archiver is improved remarkably on SSRF beamlines. A web-based GUI with a new user input method ('Markov Auto-Complete') was developed to provide better reliability and usability. Users can access the operation data easily and fast by web browser from any networked computer.

-
- [1] SSRF Home Page. <http://ssrf.sinap.ac.cn/>, September 10, 2013.
 - [2] EPICS Home Page. <http://www.aps.anl.gov/epics/>, September 10, 2013.
 - [3] Dalesio L R, Kraimer M R, Kozubal A J. EPICS architecture, ICALEPCS. 1991, **91**: 92–15.
 - [4] Kasemir K, Gabriele Carcassi. Control System Studio Guide. <http://cs-studio.sourceforge.net/docbook/>, September 10, 2013.
 - [5] JCA Home Page. <http://jca.cosylab.com/>, September 10, 2013.
 - [6] Kasemir K. Control System Studio (CSS) Data Browser, in Proc PCaPAC08, Ljubljana, Slovenia, 2008, pp. 99–101.
 - [7] del Campo M, Giacchini M, Giovannini L, *et al.* EPICS HyperArchiver: initial tests at ESSBilbao, in Proc. IPAC2011, San Sebastián, Spain, 2011, pp. 2343–2345.